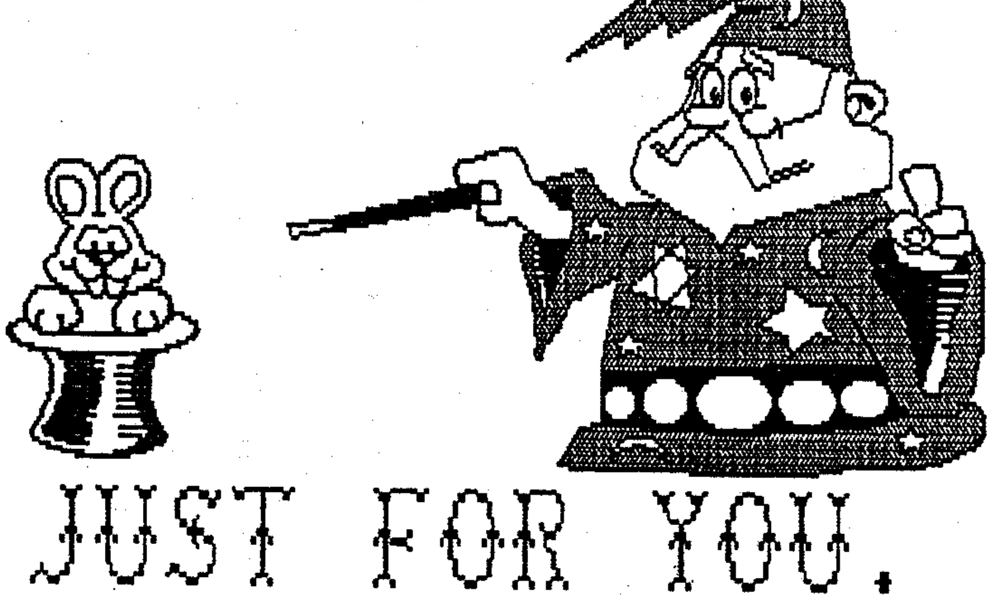
FORMAT

THE NEWSLETTER
of the
INDEPENDENT
DISCiPLE USER GROUP

ISSUE 2 - SEPTEMBER '87

DISCIPLE MAGIC



JUST FOR YOU.

INDUG.

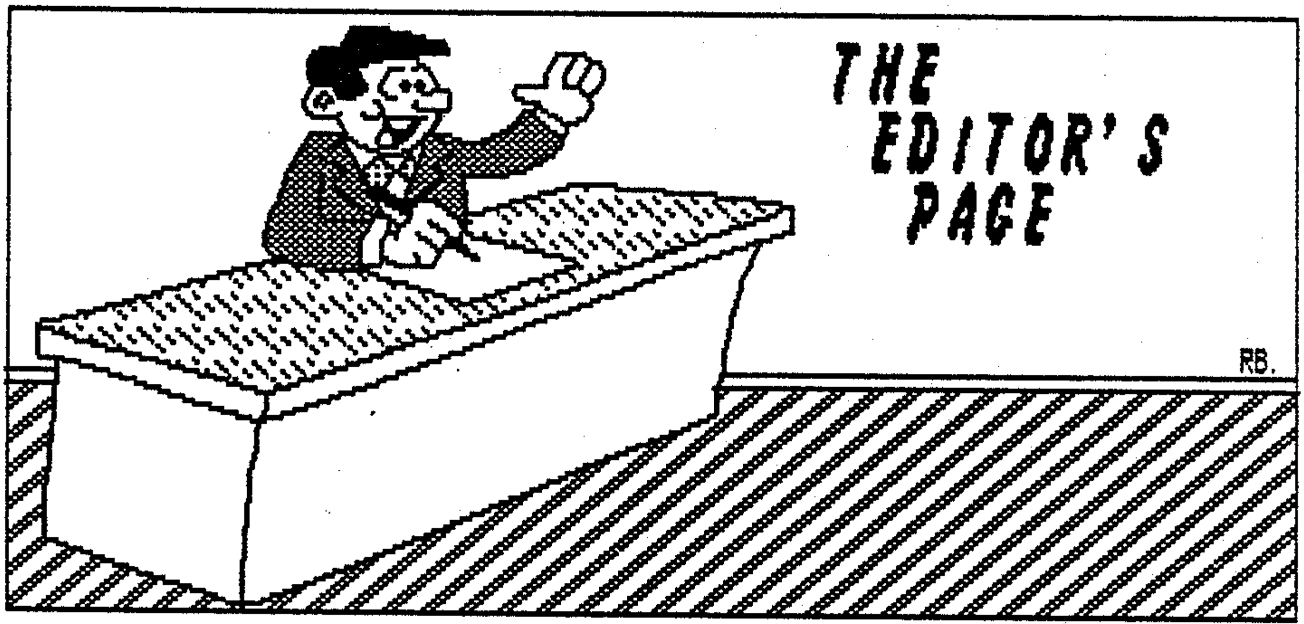
CONTENTS

ISSUE 2 - SEPTEMBER 1987

The Editors Page..............3
News..........................4
Next Month....................4
Discovery to DISCiPLE.........5
Hardware Spot.................6
Converting VU 3D..............8
POKE @ The Full Facts.........10
Program Page..................13
The Help Page.................15

FORMAT is published by:- The Independent DISCiPLE Users Group (INDUG) 34, Bourton Road, Gloucester, GL4 OLE, England.

The DISCiPLE interface is marketed by ROCKFORT PRODUCTS,
81 Church Road, London, NW4 4DP.



It's been a long, hot month since the last issue went to the printers but things have been moving very fast. Membership of INDUG is now over the 300 mark and still climbing week by week.

My DISCiPLE 'HOTLINE' has been under some pressure with users trying to work out how to make the most of their interface.

I extend a warm welcome to John Wase as an irregular contributor to these pages. John wrote the DISCiPLE page for ZX COMPUTING until its close and since has written for magazines such as YOUR COMPUTER. I trust we will see more from him in the future.

Next a call to you all for HELP, its proving impossible to keep pace with software on the DISCiPLE. I need details of DISCiPLE compatible software, instructions for conversion to disc (see this months article on VU3D) and reviews, particularly of utility and serious programs. So don't keep things to yourself, let others in on the secret.

Next month will see the start of two new services for members. First a small ads page For just £1.25 we will run your advert, up to 30 words, in the next available issue of FORMAT. If you need more space the give me a ring and we will sort something out. So if you want to SELL, BUY, EXCHANGE, ANNOUNCE or CONTACT, the small ad page in FORMAT is the ideal place.

The other new service (if you can call it a service) is a LETTERS PAGE.

This is where you can have your say about any subject close to your heart and there's an added bonus, IF your letter is really good you may win a spot prize.

If you ring and get my answer phone, please try again later, I am running INDUG on a non profit basis so I really can't return your calls. When I introduce a charge for the hotline service things may be different but in the meantime I hope you will understand.

Finally I will be away from the 28th August to the 5th September. So save your calls to I get back

Bob Brenchley.
Editor.

DISCiPLE NEWS.

ZX MICROFAIR

The ZX Microfair, at the New Horticultural Hall, Westminster, on Saturday 22nd August, was not the most well attended show on record but it did set new records for the DISCiPLE. A large area taken over by the DISCiPLE with Rockfort Products sharing the area with many other companies producing software for the DISCiPLE. A giant 45 inch T.V. promoting the 'DISCiPLE VILLAGE' dominated the entire Microfair and acted as a magnet which drew large crowds to the DISCiPLE area. At time it looked like the DISCiPLE was taking over the whole show.

The INDUG stand was staffed by Bob Brenchley, with Bruce Gordon on hand for part of the day to sort out the hardware problems. Hi-Soft, OCP, Myrmidon Software (The Last Word), Transform and Better Bytes were all showing software for the DISCiPLE, look out for reviews next month.

AMSTRAD +3 PROBLEMS

Amstrad are coming under fire from several directions over their new Spectrum+3. Hardware incompatibilities mean neither of the two existing Spectrum modems work with the new machine. It seems that both the 9v line and the NMI line are missing from the new edge connector so only the most simple hardware add-ons will now work. From the software point of view, few companies seem interested in producing disc software on the 3 inch format.

ROM LISTING

October will see the release of the official DISCiPLE ROM DISASSEMBLY. Expected price is around £15. More details next month.

NEXT MONTHS ISSUE

BUMPER ISSUE - YES we're getting bigger.

* * PROGRAMS and MORE PROGRAMS * *
Lots of useful listings to type in

* * * A **REAL** RS232 Interface * * *
Simple to build, easy to use.

* * * * **PLUS LOTS MORE** * * * *

Discovery To DISCiPLE.

By John Wase.

MOVING DISC FILES TO THE DISCiPLE

Long before the DISCiPLE came onto the market I purchased a Opus Discovery disc unit for my Spectrum. As some of you will remember, I wrote the Discovery Column for ZX Computing until its recent close. Having therefore amassed a great deal of software, files etc on Discovery disc I needed to work out an easy(ish) way to transfer them to the DISCiPLE when it arrived. There follows a short guide to solving this problem.

Although I have done this successfully using either an issue 3 48k Spectrum or an early Spectrum +2, there are one or two problems; for example, the operation seems to depend on the exact model Spectrum which you have. So although I'll go through what I did step-by-step, I give no cast iron guarantees.

Here we go. The first problem is where to connect the DISCiPLE; twixt Discovery and Spectrum, or on the Opus output port. Well, I found that for use in conjunction with Discovery, the DISCiPLE must be fitted to the Discovery output port. It is preferable, though not vital, to have the NMI line completed by soldering the connections across line 13 of the upper row of pins, but as this is particularly tricky, it should be entrusted to an electronics expert unless you feel you are specially gifted in this direction. Unless you are prepared to tilt everything up at crazy angles - bound to strain something - use a 56-way flexible edge connector to connect the DISCiPLE.

So, you've got it all connected? The next problem, and probably the most pressing of them all, is how to get the boot software in from the DISCiPLE disc. RUN and ENTER will usually get some of it in, though you might have to fiddle with the inhibit button. What usually happens is that the bulk of the systems file reads in and then the poor Spectrum jams solid. Next, you've got to unjam it without getting rid of the software, which, you will deduce, has been read into the interface, not into the Spectrum's RAM. This is where my two Spectrums differ. The +2 will not respond to "BREAK", and I therefore, have to reset it, and then do OUT 123,0 to reboot the system after resetting the +2. In contrast, the 48k Spectrum (fortunately) responds to "BREAK". I don't know what is missing from the code, but sufficient has been transferred to allow SAVES, and this is all that matters at this stage.

After this, it's all plain sailing (well, it's a bit tricky but don't worry). With the DISCiPLE's inhibit button in, do an OUT 31,0. This will permit you to catalogue the Discovery, and, by using a header-reader program (there are several around), you can find the start address and length of the code file, or the auto-line start of BASIC program. Jot these down for the files you want. Usually each program is a bit of BASIC and a couple of code files. MERGE the BASIC and enable the DISCiPLE by doing OUT 31,16. You can then adjust the syntax as necessary, before saving it to the DISCiPLE (don't forget the auto-run line number). Disable the DISCiPLE with OUT 31,0 and deal with the code (don't forget to do CLEAR (Start-1)). And so on.

It takes time and patience but it's not difficult, although you must remember to enable and disable at the appropriate times, particularly if you want to edit BASIC programs or you will be in trouble; the edit line won't be accepted back into the program. It may be better to save all files to the DISCiPLE, remove the Discovery, reconnect the DISCiPLE and then edit all the Basic programs at you leisure.

It is even possible to run the disc built into the Discovery from the DISCiPLE interface. In this case, you need a new connector; the built-in drive already has a ribbon cable, but one end is fasten by a permanent connector to the printed circuit board in Discovery. The end next to the drive is removable. Take it off, plug in the spare ribbon cable and plug the other end into DISCiPLE. The power supply on the Discovery can then be used just to power the disc drive.

A final tip or two. The messages have to be sent across at least two grotty edge connectors. Before you start, clean them with an ink eraser. And, finally, avoid any strains on the connectors. In my experience, connector trouble is responsible for most of the peripheral problems that I hear about.

And that's it: let us know how you get on.

HARDWARE SPOT

THE LIGHT STAYS ON

In the Introductory issue of FORMAT it was stated that if the light stays on, on your disc drive then this will do it no harm whatsoever. Well maybe but maybe not.

If you are using a Shugart or an MPI disc drive (5 1/4") then you could be building up problems for the future. I have no knowledge of the smaller, more modern drives, but on these there is a small shunt on the printed circuit board near to the edge connector, see FIG 1. This shunt (a dummy chip if you like) is used to set the drive address and also what signal loads the disc heads. As can be seen from FIG 2. there are two ways of loading the head either when the drive is selected (short pins

1-14) or when the motor is running (short pins 7-8).

The problems that you can get are caused by leaving the heads loaded when the disc is not moving. This is the case when using drive select. The problems are:-

Rapid wear of the disc when it stops and starts with the heads loaded. This can cause read errors on the disc, but more important the oxide that is removed can damage the heads. And the damaged heads cause more rapid wear of the discs.

A depression in the disc where the heads have been resting on the surface, causing read error at that point If the power is removed with the heads loaded (something that you must never do with a disc in anyway) then you will almost certainly lose that sector. The disc will be OK after reformatting.

The solution is to select head load with motor on.

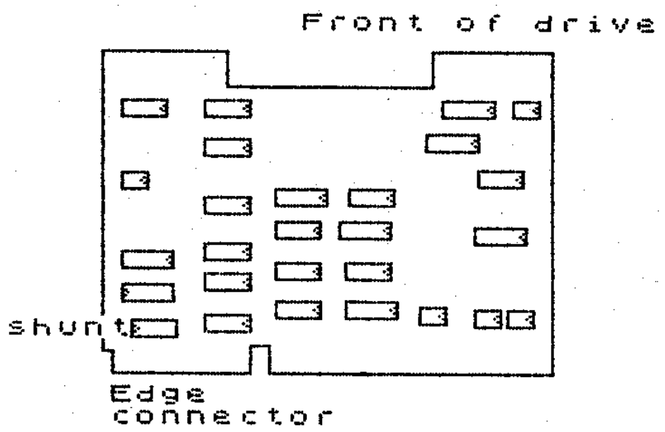


FIG 1.

PINS	ACTION
1 - 14	HEAD LOAD WITH DRIVE SELECT
2 - 13	DRIVE ADDR = 1
3 - 12	DRIVE ADDR = 2
4 - 11	DRIVE ADDR = 3
5 - 10	MUX (DRIVE ALWAYS SELECTED)
6 - 9	DRIVE ADDR = 4
7 - 8	HEAD LOAD WITH MOTOR ON

FIG 2.

VU-3D
A CASE FOR CONVERSION

VU-3D is a program, written in 1982 by Psion, and published on tape by Sinclair Research. Most people who purchased their Spectrum with the free Sinclair six-pack will have found VU-3D included. It allows simple three dimensional objects to be drawn on the screen. Once drawn an object can be:

saved.
enlarged or reduced in size.
shown in perspective at varying distances.
rotated about two axes.
have hidden lines removed.
illuminated from any one of six directions,
 (this gives surface shading but no shadows)
coloured
saved as a SCREEN$ file.

There are however some disadvantages:

only a limited range of shape can be drawn.
editing facilities are very limited.
the program takes a very long time to load from tape.
cursor movement and figure rotation are painfully slow on the
 Spectrum+ and later machines.

The third of these can be overcome by converting the program to disc on the DISCiPLE, following the steps in this article. Snapshotting is no good in this case as the program would still use the tape system for its files. The last disadvantage can be overcome by making small changes to the machine code part of the program, as will be described later.

LOADing from cassette

VU-3D on tape consists of four files, two Basic programs and two Code files.

"VU3D" is a two line Basic loader program.
"vu3d" is a long Basic program which interacts with the user.
"s" is a SCREEN$ file giving something to look at while the main code file loads in.
"c" is the main machine code which does all the hard number crunching. The first few bytes of this file contain the User Defined Graphics used by the program.

In practice, neither the loader program nor the SCREEN$ file are required on the disc system. We start the transfer by doing a

LOAD "vu3d", allowing the program to load and then 'BREAKing' into the program before it starts to load the next file. Now type LOAD "c"CODE to load the machine code in the last file.

PROGRAM MODIFICATIONS

Once the Basic and Code have been loaded they require modification before saving to disc.
The SAVE and LOAD instructions need to have d1 or d2 added to select the disc, these instructions occur at lines 630, 1110, 2700 and 5120.
Lines 4999 and 5100 are LOAD "" SCREEN$ instructions but as there seems to be no call to these lines there is no need to modify them.

Two useful additions to the program are:

2055 IF e$="8" THEN STOP. keying 8 while in the main menu will then exit the program (restart by GOTO 1).

Insert CAT 1 at the beginning of line 1110. This puts the disc directory on the screen whenever it is desired to load a file.

As I said at the beginning of this article the cursor movement is very slow. This is because the program was intended for the old 48k Spectrum. Keys 5,6,7 and 8 are used for normal (fast) cursor movement. Pressing Caps Shift and any of these keys gives a slow movement. On the Spectrum+, 128 or +2 the cursor (arrow) keys automatically activates Caps Shift, so the movement is always slow. The following modification to the machine code part of the program allows full speed cursor movement and figure rotation, and enables the Symbol Shift key to be used for slow movement.

```
POKE 45229,127
POKE 45233,79
POKE 50812,127
POKE 50816,79
```

Since Symbol Shift is now used instead of Caps Shift, lines 220 and 400 should be edited to show this change on the screen.
The entry point of "vu3d" is line 5110 so save the Basic program by

```
SAVE d1"vu3d" LINE 5110
```

then save the code file by

```
SAVE "c" CODE 31501,32183.
```

To run the program simply type LOAD d1"vu3d" and press enter.

This gives a new lease of life to an old favourite and I hope this will show readers that there are other ways too transfer a program than by using the Snapshot button.

EH Cooke-Yarborough.

POKE @

WHAT THE MANUAL LEFT OUT...

Following on from the extensive look at OPEN# & CLOSE# last month lets take a detailed look at another command in GDOS version 3.

POKE @ is a command that has been around since version one of the DISCiPLE. It has always been used to set the DISCiPLE's system variables and controls (more on this later) but it does have another use in version 3.

POKE @ has a base address of 664 (298 hex) in V3, so POKE @ 0,1 would set location 664 in GDOS to 1. POKE @ 1 would be location 665 and so on. Fine, but not a lot of use if you don't make regular changes to your system. Well how about POKE @ 15720,255. Yes the addresses are not limited to the DISCiPLE's DOS area, 15720 is the first byte of the SCREEN area (16384 - 664 =15720) so POKEing it with 255 will give a line at the top left of the screen.

Nice, but you can put things on the screen at any time with the normal Spectrum version of POKE so what makes POKE @ so useful? 65535 - thats what. If the value to be POKEed is over 255 then a double length POKE @ is done. So POKE @ 15720,43690 will but a dotted line, 2 characters wide, on the screen in the top left hand corner. POKE @ 23011,50000 would set the Spectrum system variable 'UDG' to 50000. NO MORE WORKING OUT TO GET THE TWO BYTES, LET THE DISCiPLE DO THE WORK.

This double length POKE @ will only work with values over 255 but you could force GDOS to do it with values of 0-255 if you do a POKE @ 444,1110 first. This changes the logic of the POKE @ routine so to reset it to normal use POKE @ 444,1049. On some computers this double length poke is called a DOKE.

On the Subject of POKE @ lets consider the DISCiPLE's system variables. A list of those in version 2c is given in the manual but two have changed in V3 and some new ones have been added so lets go through the list.

POKE @ 0,n (n= 0 to 7) default 7.

If 0 then the screen border remains unchanged. Any other value will cause the border to flash when reading or writing to the disc.

POKE @ 1,n (n= 0,40,80,168 or 208) default 208
POKE @ 2,n (n= 0,40,80,168 or 208) default 208

These two variables control the number of tracks on your disc drive(s). 0 = no drive, but the system configuration program insists you have a number one drive. 40 or 80 are the normal and if your drives are double sided then 128 is added. The default is therefore 80 track double sided.

POKE @ 3,n (n=6 to 255) default 6

When a drive's head moves from track to track there must be a short delay before you read or write data on the disc. This variable controls the length of this delay (called the STEP RATE) in milliseconds. It is not allowed to go below 6ms and with older drives you may need to use 12, 24, 36 or even 48ms to avoid errors.

POKE @ 4,n (1 to 64) default 1

This sets the Network Station Number. 0 = No network, 1 = Master Station, 2-9 = Assistant Stations, 10-63 = Pupil Stations.

POKE @ 5,n (n= 0 to 255) default 80

The line length for LPRINT and LLIST.

POKE @ 6,n (n=0 or >0) default 0

When set to 0 the DISCiPLE will filter out all codes < 32 (Space) except the Carriage Return, code 13. It converts Basic Tokens to their expanded form and converts TAB and PRINT AT into line feeds and spaces. If set to any other value (normally 1), then any codes sent to the printer are sent unaltered. This allows printer control codes to be sent. Before doing an LLIST ensure POKE @ 6,0 is in force or your listing will not come out as expected.

POKE @ 7,n (n=0 to 255) default 12

Line spacing. This variable controls the n/72 line increments the printer does. It is used by the DISCiPLE to reset the line spacing after printing graphics.

POKE @ 8,n (n=0 to 255) default 1

The number of line feeds (ASCII 10) that the DISCiPLE will send after each Carriage Return (ASCII 13). If your printer does an auto <LF> after <CR> then set this to 0.

POKE @ 9,n (n=0 to 255) default 0

Left margin, the number of spaces printed by the DISCiPLE before the first character on a line.

POKE @ 10,n (n=0 or 1) default 1

If your printer allows Hi-Res graphics this variable, when set to 1, will let the DISCiPLE print the £,# and (c) signs. Each prints in an 8X8 pixel form.

POKE @ 11,n (n=0 or 1) default 0

If 0 then the printer channel (#3) is set to the DISCiPLE's printer. If 1 the channel remains unaltered. If set to 0 when you BOOT your system, POKEing to 1 will not reset to the Spectrum's default (ZX on the 48k and RS232 on the 128k).

POKE @ 12 & 13 RESERVED.

POKE @ 14 & 15 (n=256 to 65535) default 0

These two bytes, if non zero, cause a CALL to your machine code routine to extend Basic. i.e. POKE @ 14,49152 will cause the DISCiPLE to jump to a routine at 49152 every time it finds a Basic statement it can't handle.

POKE @ 16 & 17 (n=256 to 65535) default 0

These two bytes, if non zero, cause a CALL to your routine every time the Spectrum scans the keyboard (50 time a second if interrupts are enabled). You could use this for a timer or to update a screen display.

We now come to the area holding the printer control codes. NOTE: All printer codes should be terminated by a character 128 if less than 8 codes are used.

POKE @ 18 to 25 Printer Initialise. Default 27,64

POKE @ 26 to 33 Character Pitch. Default 27,77

POKE @ 34 to 41 Line Spacing Set. Default 27,65

POKE @ 42 to 49 Set Graphic Code. Default 27,42,5

POKE @ 50 to 57 Any Other Codes (Sent on Init.) Default None

There now follow the bytes for the Hi-Res Characters.

POKE @ 58 to 65 Codes for £

POKE @ 66 to 73 Codes for #

POKE @ 74 to 81 Codes for (c)

POKE @ 82 to 89 Set Graphics (A4) dump. Default 27,42,5,64,2

POKE @ 90 to 98 Gray Scale Table.

A table used by the A4 screen dump routine to decide the dot pattern for each of the eight colours on a Spectrum screen (Bright & Flash are ignored). Each byte provides 1 bit for the pattern. Each screen pixel gives a 3X3 printer pattern hence 9 bytes in table.

Version 2c users have the same system variables up to POKE @81 (the end of the (c) sign bytes) except that POKE @ 16 & 17 are not used. There is however one other small difference, POKE @ 4 and POKE @ 11 swap over their use.

POKE @ is, I think you will agree, a powerful command. Lets see a few of you think up new uses for the double length POKE, our pages are open for your ideas so get your thinking caps on.

Bob Brenchley.

*** SUPER - CAT ***

This program gives a slightly altered disc catalogue to a printer. The program is written for the new version 3 DISCiPLE and uses the new LOAD sector command. The program is set up for a double density disc system where each sector is 512 bytes long, single density users will have to alter the program listing as necessary.

The program first loads the directory tracks into memory, four tracks each of ten sectors and each 512 bytes long. Each file header is 256 bytes long so that all 80 possible files are stored in 20K of disc space.

When loaded into memory you can use the memory print routine to view each header and you will see that very little of the space is used. The first byte of the header is the file type - see line 150 of the program listing for the 11 possible types. If this is 0 then the header space is unused and is available to be over-written by the next SAVE. If you accidentally ERASE a file all the sectors are still present, unchanged on the disc and the header is also present with one alteration, the first byte is changed to a 0, meaning the header will not be listed by the DOS CATalogue command and therefore, it can not be loaded. However, if you repair the damage, before the next SAVE or OPEN, by loading the directory sector into memory altering the first byte to the file type and resaving you will have restored the file.

I have also listed the file track and sector in my catalogue listing so that you can trace all the sectors used on the disc for each of the files if you wish. The first sector of each program file consists of 9 header bytes as copied from the directory sector. The next 501 bytes are the file data and the final two bytes point to the next track and sector which need not necessarily be consecutive. This and all subsequent sectors consist of 510 data bytes and the last two bytes pointing to the next sector or holding a value of 0 if there are no further sectors to load.

The listing is written in simple BASIC, so it should be easy to follow. The values required are extracted from each header and listed. Byte 1 is the file type, Bytes 2 to 11 are the file name, Byte 12 is the number of sectors used for the file, Bytes 13/14 are pointers to the first track/sector I have been unable to decode the next bytes, they are presumably some check sum values although altering them and resaving the header does not seem to have any effect, the programs still load. Byte 211 is the file type - see page 57 of the Disciple manual. Bytes 212/3 give the file length and Bytes 214/5 the code start position or 23755 (5CCB Hex) for program files. Bytes 218/9 are either the BASIC autostart line or the CODE autostart location. If they hold a value of FFFFh (or 0 for a CODE file) then they are unused.

As the LOAD @ command is used the program will only work on DISCiPLES fitted with version 3 ROMs. If this was replaced with a short machine code routine to read the sectors then it should be possible to get it working on V2 systems.

David Kennedy. London.

```
   5 POKE @6,1: CLS : PRINT  INVERSE 1;AT 1,10;"CHOICE  MENU"; I
NVERSE 0'''"'C'  DISC CATALOGUE"'''"'S'  SECTOR PRINT"'''"'M'  MEM
ORY PRINT": PAUSE 0: CLS : LET A=PEEK 23556: GO TO (200 AND A=CO
DE "S")+(240 AND A=CODE "M")+(10 AND A=CODE "C")
   6 RUN
  10 LET m=30000: FOR t=0 TO 3: FOR s=1 TO 10: LOAD @1,t,s,m: LE
T m=m+512: PRINT AT 6,2;"TRACK ";T;" SECTOR ";S;" ": NEXT s: NEX
T t: PRINT ''"  DIRECTORY LOADED"
  20 LPRINT CHR$ 27;CHR$ 48: LET TOT=0: INPUT "DISC NUMBER ";D'"
DATE ";A$: IF LEN A$>13 THEN  GO TO 20
  25 LPRINT CHR$ 14;CHR$ 27;CHR$ 45;CHR$ 1;"DISK NUMBER: ";D;CHR
$ 27;CHR$ 45;CHR$ 0;"     Date: ";A$''
  30 POKE @6,0: RESTORE 150: FOR T=1 TO 11: READ A$: LET NL=0: F
OR A=0 TO 79: LET B=PEEK (30000+A*256): IF B=T THEN  GO SUB 100
  40 LET TOT=TOT+PEEK (30012+A*256): NEXT A: IF NL THEN  LPRINT
  50 NEXT T: LPRINT : LPRINT : LPRINT "  FREE SPACE: ";INT (780-
TOT/22);" K  BYTES": LPRINT : LPRINT : STOP
 100 LPRINT A+1;(" " AND A<9);"  ";: FOR C=30001+A*256 TO 30010+
A*256: LPRINT CHR$ (PEEK C);: NEXT C: LPRINT "   ";
 110 LPRINT A$;: IF B=4 THEN  LPRINT PEEK (30214+A*256)+256*PEEK
 (30215+A*256);",";PEEK (30212+A*256)+256*PEEK (30213+A*256);: L
ET C=PEEK (30218+A*256)+256*PEEK (30219+A*256): IF C<>65535 THEN
  IF C THEN  LPRINT ", run ";C;
 120 IF B=1 THEN  LET C=PEEK (30218+A*256)+256*PEEK (30219+A*256
): IF C<>65535 THEN  LPRINT "LINE ";C;
 130 LET NL=NL+1: LPRINT TAB 50;"(Len: ";PEEK (30012+256*a);")";
TAB 63;"Tr ";PEEK (30013+256*A);TAB 71;"Sct ";PEEK (30014+256*a)
: RETURN
 150 DATA "BASIC     ","N. ARRAY","$ ARRAY","CODE      ","SNAPSHOT 4
8K","M/DRIVE FILE","SCREEN$","SPECIAL FILE","SNAPSHOT 128K","OPE
N# TYPE","EXECUTE"
 200 LET M=30000: INPUT "LOAD TRACK NO. ";T''"SECTOR NO. ";S
 210 LOAD @1,T,S,M
 220 PRINT AT 10,0;"Track ";T;''"Sector ";S;" "''"Mem Loc. ";M:
LET M=M+512: LET S=S+1: IF S=11 THEN  LET S=1: LET T=T+1
 230 PRINT ''"ENTER TO REPEAT": PAUSE 0: IF PEEK 23560=13 THEN
GO TO 210
 240 CLS : PRINT "SECTOR LINE PRINT": INPUT "PRINT FROM ";M
 250 FOR A=0 TO 511 STEP 16: LET A$="": LPRINT A;"  ";(" " AND A
<99)+(" " AND A<9);
 260 FOR B=0 TO 15: IF PEEK (M+A+B)=0 THEN  NEXT B: LPRINT "00":
 NEXT A: GO TO 280
 270 FOR B=0 TO 15: LET C=PEEK (M+A+B): GO SUB 300: NEXT B: LPRI
NT "     ";A$: NEXT A
 280 LET M=M+512: LPRINT : LPRINT : PRINT "ENTER FOR NEXT SECTOR
": PAUSE 0: IF INKEY$=CHR$ 13 THEN  GO TO 250
 290 POKE @6,0: STOP
 300 LET D=INT (C/16): LET E=C-D*16: LPRINT CHR$ (D+48+7*(D>9));
CHR$ (E+48+7*(E>9));" ";: IF C<32 OR C>127 THEN  LET C=CODE "."
 310 LET A$=A$+CHR$ C: RETURN
```

THE HELP PAGE

Problems with your DISCiPLE? Dont worry, write to the HELP page Remember to quote your membership number and leave the problem to us.

COPY ERROR

I have just fitted the version 3 upgrade to my DISCiPLE. With V2 COPY d1"xxxx" TO d1 worked but with GDOS-3 I keep getting a "Syntax Error" report. The extra sheets supplied with the upgrade say nothing about this. Am I doing something wrong or is the EPROM faulty.

David Fullerton. Bristol.

Its O.K. David, the fault lies in the Rockfort printed sheets which failed to tell you of the change (needed to get round a bug in the 128's ROM) made to the copy command. Just replace COPY with SAVE, the syntax stays the same. This also applies to COPY SCREEN$ which now becomes SAVE SCREEN$ although in 48k mode COPY SCREEN$ still appears to work.

HANGING BOOT

When I 'Boot' my system, with RUN and ENTER, the 'Miles Gordon Technology DOS 3b' message appears but then nothing happens. I have to press BREAK to continue. Is there a bug?

R.McLean. South Africa.

If you set up your system to use the DISCiPLE's printer port and then 'boot' with the printer disconnected or off-line you will get this problem. Connect the printer,with some printer they can even be switched off, or put the printer on-line. Everything should then work fine.

NETWORK SNAPSHOTS

I have been trying to use some programs over a small Network (3 Spectrums) and I am having problems. The programs are SNAPSHOTs on disc and load/run fine on the master station. How do I get them over to the other stations.

John Sinclair. Glasgow.

Sorry but there's no way to transfer SNAPSHOT files over the Net. This is partly as a concession to software companies, especially the educational ones. It is also due to the size of a SNAPSHOT file, at a full 48k there is no room in the master computer to SEND the program.

Thats all for this month. keep those problems coming in.

VIDEO or FILM?

WHY NOT JOIN US

IAC for FILM, VIDEO, and A/V
We offer the following services

*Amateur Film Maker ..... a magazine for enthusiasts
*Film Library
*Insurance Cover
*Copyright Clearance
*Regional Events
*Mood Music Service
*Film and Video Competitions
*Taped Lecture Service
*Technical Advice
*Music Advisory Service
*National Events
*Contact with other enthusiasts at home and abroad

For further details send a SAE to:-
IAC Dept 'P' 63 Woodfield Lane ASHTEAD Surrey
KT21 2BT or 'Phone 03722-76358